

# QSM368ZP-WF&SG368Z 系列

## Linux&Ubuntu&OpenWrt

### 分区调整指导

智能模块系列

版本：1.1

日期：2024-09-29

状态：受控文件



上海移远通信技术股份有限公司（以下简称“移远通信”）始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司  
上海市闵行区田林路 1016 号科技绿洲 3 期（B 区）5 号楼 邮编：200233  
电话：+86 21 5108 6236 邮箱：[info@quectel.com](mailto:info@quectel.com)

或联系我司当地办事处，详情请登录：<http://www.quectel.com/cn/support/sales.htm>。

如需技术支持或反馈我司技术文档中的问题，请随时登录网址：  
<http://www.quectel.com/cn/support/technical.htm> 或发送邮件至：[support@quectel.com](mailto:support@quectel.com)。

## 前言

移远通信提供该文档内容以支持客户的产品设计。客户须按照文档中提供的规范、参数来设计产品。同时，您理解并同意，移远通信提供的参考设计仅作为示例。您同意在设计您目标产品时使用您独立的分析、评估和判断。在使用本文档所指导的任何硬软件或服务之前，请仔细阅读本声明。您在此承认并同意，尽管移远通信采取了商业范围内的合理努力来提供尽可能好的体验，但本文档和其所涉及服务是在“可用”基础上提供给您的。移远通信可在未事先通知的情况下，自行决定随时增加、修改或重述本文档。

## 使用和披露限制

### 许可协议

除非移远通信特别授权，否则我司所提供硬软件、材料和文档的接收方须对接收的内容保密，不得将其用于除本项目的实施与开展以外的任何其他目的。

### 版权声明

移远通信产品和本协议项下的第三方产品可能包含受移远通信或第三方材料、硬软件和文档版权保护的相关资料。除非事先得到书面同意，否则您不得获取、使用、向第三方披露我司所提供的文档和信息，或对此类受版权保护的资料进行复制、转载、抄袭、出版、展示、翻译、分发、合并、修改，或创造其衍生作品。移远通信或第三方对受版权保护的资料拥有专有权，不授予或转让任何专利、版权、商标或服务商标权的许可。为避免歧义，除了正常的非独家、免版税的产品使用许可，任何形式的购买都不可被视为授予许可。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法侵权行为，移远通信有权追究法律责任。

### 商标

除另行规定，本文档中的任何内容均不授予在广告、宣传或其他方面使用移远通信或第三方的任何商标、商号及名称，或其缩略语，或其仿冒品的权利。

### 第三方权利

您理解本文档可能涉及一个或多个属于第三方的硬软件和文档（“第三方材料”）。您对此类第三方材料的使用应受本文档的所有限制和义务约束。

移远通信针对第三方材料不做任何明示或暗示的保证或陈述，包括但不限于任何暗示或法定的适销性或特定用途的适用性、平静受益权、系统集成、信息准确性以及与许可技术或被许可人使用许可技术相关的不侵犯任何第三方知识产权的保证。本协议中的任何内容都不构成移远通信对任何移远通信产品或任何其他硬件、设备、工具、信息或产品的开发、增强、修改、分销、营销、销售、提供销售或以其他方式维持生产的陈述或保证。此外，移远通信免除因交易过程、使用或贸易而产生的任何和所有保证。

## 隐私声明

为实现移远通信产品功能，特定设备数据将会上传至移远通信或第三方服务器（包括运营商、芯片供应商或您指定的服务器）。移远通信严格遵守相关法律法规，仅为实现产品功能之目的或在适用法律允许的情况下保留、使用、披露或以其他方式处理相关数据。当您与第三方进行数据交互前，请自行了解其隐私保护和数据安全政策。

## 免责声明

- 1) 移远通信不承担任何因未能遵守有关操作或设计规范而造成损害的责任。
- 2) 移远通信不承担因本文档中的任何因不准确、遗漏、或使用本文档中的信息而产生的任何责任。
- 3) 移远通信尽力确保开发中功能的完整性、准确性、及时性，但不排除上述功能错误或遗漏的可能。除非另有协议规定，否则移远通信对开发中功能的使用不做任何暗示或法定的保证。在适用法律允许的最大范围内，移远通信不对任何因使用开发中功能而遭受的损害承担责任，无论此类损害是否可以预见。
- 4) 移远通信对第三方网站及第三方资源的信息、内容、广告、商业报价、产品、服务和材料的可访问性、安全性、准确性、可用性、合法性和完整性不承担任何法律责任。

版权所有 © 上海移远通信技术股份有限公司 2024，保留一切权利。

**Copyright © Quectel Wireless Solutions Co., Ltd. 2024.**

# 文档历史

## 修订记录

版本	日期	作者	变更描述
-	2023-11-29	Szymon YU	文档创建
1.0	2024-03-08	Szymon YU	受控版本
1.1	2024-09-29	Ivan Zujovic	<ol style="list-style-type: none"> <li>更新文档名，新增“OpenWrt”字样</li> <li>新增 OpenWrt 系统说明（第 1 章）</li> <li>新增 OpenWrt 系统模块分区表路径及 parameter_tool.exe 工具路径（第 3.1.1 章和第 3.1.2 章）</li> <li>新增 OpenWrt 系统分区镜像创建及分区添加结果验证操作指导（第 4.2.2 章和第 4.3.2 章）</li> </ol>

## 目录

文档历史 .....	3
目录 .....	4
图片索引 .....	5
1 引言 .....	6
2 分区调整原则.....	7
3 分区修改.....	8
3.1. 物理分区修改.....	8
3.1.1. 分区表位置 .....	8
3.1.2. 通过工具修改分区 .....	8
3.2. Linux/Ubuntu 分区镜像修改 .....	12
3.2.1. 根文件系统分区镜像修改（仅适用于 Linux） .....	13
3.2.2. 其他分区镜像修改 .....	13
3.3. 分区修改结果验证 .....	14
4 分区添加.....	15
4.1. 物理分区添加.....	15
4.2. 分区镜像创建.....	16
4.2.1. Linux/Ubuntu 分区镜像创建 .....	16
4.2.2. OpenWrt 分区镜像创建 .....	18
4.3. 分区添加结果验证 .....	21
4.3.1. Linux/Ubuntu 分区添加结果验证 .....	21
4.3.2. OpenWrt 分区添加结果验证 .....	22

## 图片索引

图 1: parameter_tool.exe 工具主界面 .....	9
图 2: 选择分区文件.....	10
图 3: 分区参数列表.....	10
图 4: 选择分区 .....	11
图 5: 修改 oem 分区信息.....	11
图 6: 确认并保存修改 .....	12
图 7: 分区修改成功.....	12
图 8: 根文件系统分区镜像大小修改位置.....	13
图 9: 其他分区镜像大小修改位置.....	13
图 10: 分区修改成功.....	14
图 11: 插入新物理分区 .....	15
图 12: 设置新物理分区大小和名称.....	16
图 13: 分区添加成功.....	16
图 14: 添加分区镜像配置 .....	17
图 15: 添加分区文件.....	17
图 16: 添加配置创建分区镜像 .....	17
图 17: 添加开机自动挂载配置 .....	18
图 18: 添加 new 文件夹.....	18
图 19: 创建 normal 文件夹 .....	19
图 20: 添加新分配配置 .....	19
图 21: 进入 Extra partitions 子菜单 .....	19
图 22: 修改 number of extra partitions .....	20
图 23: 修改分区配置.....	20
图 24: 保存配置.....	21
图 25: 检查 output/.config 新分区文件内容 .....	21
图 26: 添加物理分区成功 .....	22
图 27: 分区镜像文件添加成功 .....	22
图 28: 使用 RKDevTool 工具烧录固件 .....	23
图 29: OpenWrt 环境.....	23
图 30: 检查挂载点 .....	23
图 31: new 分区文件内容 .....	24

# 1 引言

本文档主要介绍如何在移远通信 QSM368ZP-WF 和 SG368Z 系列模块中调整 eMMC (Embedded Multi Media Card, 嵌入式多媒体卡) 分区, 包括分区修改和分区添加相关步骤。

## 备注

1. 本文适用于运行 Linux 操作系统的 QSM368ZP-WF 和运行 Linux 或 Ubuntu 操作系统的 SG368Z 系列模块。
2. SG368Z 系列模块本身无法预装 OpenWrt 系统, 仅提供 SDK 及开发指导供客户二次开发。若有问题, 请联系移远通信技术支持。

## 2 分区调整原则

1. 仅可在所有其他分区之后、*userdata* 分区之前添加分区。
2. 可修改分区表中任意分区的大小。不建议修改 *recovery*、*persist*、*nvdata1* 和 *nvdata2* 分区。
3. 不建议修改原生分区的名称。



## 3 分区修改

### 3.1. 物理分区修改

在 Windows 系统中，使用 *parameter\_tool.exe* 工具修改分区表相关信息实现对物理分区的修改，具体如下。

#### 3.1.1. 分区表位置

QSM368ZP-WF 和 SG368Z 系列模块的分区表为 *parameter-buildroot-fit.txt*，位于如下源码路径：

- Linux: *rk3568\_linux\_r60\_v1.3.2/device/rockchip/rk356x*
- Ubuntu: *rk3568\_ubuntu\_r60\_v1.3.2/device/rockchip/rk356x*
- OpenWrt: *rk3568\_openwrt\_r61\_v1.4.0/device/rockchip/rk356x*

#### 3.1.2. 通过工具修改分区

在 Windows 系统中，使用 *parameter\_tool.exe* 工具修改分区大小和名称。该工具的压缩包位于如下路径：

- Linux: *rk3568\_linux\_r60\_v1.3.2/tools/windows/ParameterTool\_v1.2.zip*
- Ubuntu: *rk3568\_ubuntu\_r60\_v1.3.2/tools/windows/ParameterTool\_v1.2.zip*
- OpenWrt: *rk3568\_openwrt\_r61\_v1.4.0/tools/windows/ParameterTool\_v1.2.zip*



图 1: parameter\_tool.exe 工具主界面

物理分区大小和名称修改步骤如下：

**步骤 1：** 单击“**Browse**”按钮。

**步骤 2：** 在项目代码中找到 *parameter-buildroot-fit.txt* 文件的位置。

**步骤 3：** 选择 *parameter-buildroot-fit.txt* 文件，如下图所示：

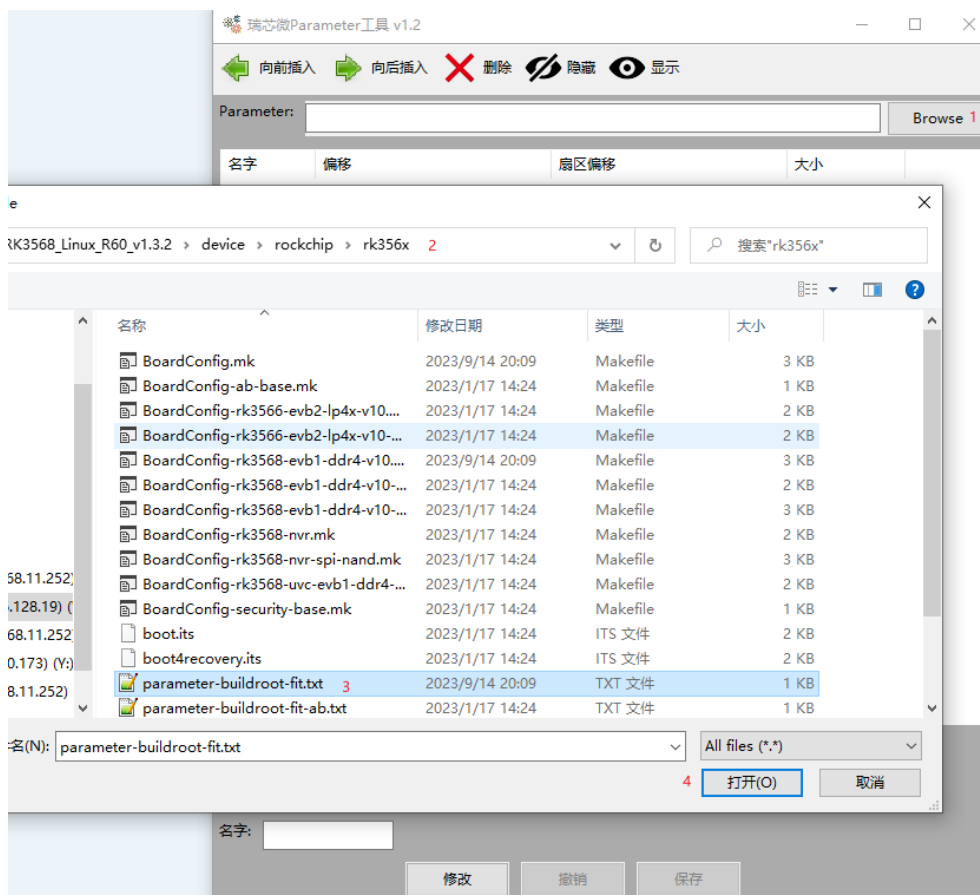


图 2：选择分区文件

步骤 4：打开文件后，将会看到分区参数列表，如下图所示：



图 3：分区参数列表

步骤 5: 在分区参数列表中, 选择需要修改的分区, 分区信息如下图所示:

Parameter: W:\RK3568\_Linux\_R60\_v1.3.2\device\rockchip\rk356x\parameter-buildroot-fit.txt Browse

名字	偏移	扇区偏移	大小
uboot	0x800000	0x4000	4MB
misc	0xC00000	0x6000	4MB
boot	0x1000000	0x8000	64MB
recovery	0x5000000	0x28000	64MB
persist	0x9000000	0x48000	32MB
nvdata1	0xB000000	0x58000	4MB
nvdata2	0xB400000	0x5A000	4MB
backup	0xB800000	0x5C000	32MB
rootfs	0xD800000	0x6C000	6144MB
oem	0x18D800000	0xC6C000	128MB
userdata:g...	0x195800000	0xCAC000	-

偏移: 0x18D800000

大小: 128 ☐ KB ☒ MB

名字: oem

修改 撤销 保存

图 4: 选择分区

步骤 6: 修改分区名称和大小。以 *oem* 分区为例, 输入需要修改的分区大小和名称。例如, 将 *oem* 分区大小修改为“256 MB”, 并修改分区名称为“oem\_test”, 如下图所示:

Parameter: W:\RK3568\_Linux\_R60\_v1.3.2\device\rockchip\rk356x\parameter-buildroot-fit.txt Browse

名字	偏移	扇区偏移	大小
uboot	0x800000	0x4000	4MB
misc	0xC00000	0x6000	4MB
boot	0x1000000	0x8000	64MB
recovery	0x5000000	0x28000	64MB
persist	0x9000000	0x48000	32MB
nvdata1	0xB000000	0x58000	4MB
nvdata2	0xB400000	0x5A000	4MB
backup	0xB800000	0x5C000	32MB
rootfs	0xD800000	0x6C000	6144MB
oem	0x18D800000	0xC6C000	128MB
userdata:g...	0x195800000	0xCAC000	-

偏移: 0x18D800000

大小: 256 ☐ KB ☒ MB

名字: oem\_test

修改 撤销 保存

图 5: 修改 oem 分区信息

**步骤 7:** 设置完成后，点击下图中的“修改”按钮，确认无误后，单击“保存”按钮，修改即可生效。如需取消修改，单击“撤销”按钮。指定分区修改完成后，被修改分区和后续分区的大小偏移值也会随之改变。

名字	偏移	扇区偏移	大小
uboot	0x800000	0x4000	4MB
misc	0xC00000	0x6000	4MB
boot	0x1000000	0x8000	64MB
recovery	0x5000000	0x28000	64MB
persist	0x9000000	0x48000	32MB
nvdata1	0xB000000	0x58000	4MB
nvdata2	0xB400000	0x5A000	4MB
backup	0xB800000	0x5C000	32MB
rootfs	0xD800000	0x6C000	6144MB
oem_test	0x18D80000	0xC6C000	256MB
userdata:grow	0x19D800000	0xCEC000	-

偏移: 0x18D80000  
 大小: 256 ☐ KB ☒ MB  
 名字: oem\_test

图 6：确认并保存修改

**步骤 8:** 检查 *parameter-buildroot-fit.txt* 文件，将会看到分区信息已被修改，如下图所示：

```
--- a/device/rockchip/rk356x/parameter-buildroot-fit.txt
+++ b/device/rockchip/rk356x/parameter-buildroot-fit.txt
@@ -8,5 +8,5 @@ MACHINE: 0xffffffff
CHECK_MASK: 0x80
PWR_HLD: 0,0,A,0,1
TYPE: GPT
-CDLINE: mtdparts=rk29xxnand:0x00020000@0x00040000(uboot),0x00020000@0x00060000(misc),0x00020000@0x00080000(boot),0x00020000@0x000A0000(recovery),0x00010000@0x000C0000(persist),0x00020000@0x000E0000(nvdata1),0x00020000@0x000F0000(nvdata2),0x00010000@0x00100000(backup),0x000C0000@0x00140000(rootfs),0x00040000@0x00160000(oem),-@0x00C00000(userdata:grow)
+CDLINE: mtdparts=rk29xxnand:0x00020000@0x00040000(uboot),0x00020000@0x00060000(misc),0x00020000@0x00080000(boot),0x00020000@0x000A0000(recovery),0x00010000@0x000C0000(persist),0x00020000@0x000E0000(nvdata1),0x00020000@0x000F0000(nvdata2),0x00010000@0x00100000(backup),0x000C0000@0x00140000(rootfs),0x00040000@0x00160000(oem_test),@0x00CEC00000(userdata:grow)
uuid:rootfs=614e0000-0000-4b53-8000-1d28000054a9
```

图 7：分区修改成功

## 3.2. Linux/Ubuntu 分区镜像修改

对于 Linux/Ubuntu 的分区镜像，如 *recovery*（不建议修改）、*rootfs*、*oem* 等，完成第 3.1 章分区表修改后，还需要在 Linux/Ubuntu 项目代码中设置相应的配置项，本章节介绍如何修改不同分区的配置项。

### 备注

Ubuntu 的 *rootfs* 不需要在项目代码中修改。本章节介绍的 *rootfs* 修改仅适用于 Linux。

### 3.2.1. 根文件系统分区镜像修改（仅适用于 Linux）

*recovery* 和 *rootfs* 文件系统由项目代码直接编译，与这些文件系统相关的分区可以在代码中进行配置，Linux 默认设置为 *AUTO*，表示自适应。也可以自行配置，通过 *buildroot/configs/rockchip/base.config* 文件中的编译参数 *BR2\_TARGET\_ROOTFS\_EXT2\_SIZE* 进行配置，将分区镜像大小配置为指定值。需要注意的是，随着项目的进展，根文件系统会变得越来越大会越来越大，可能会引起根文件系统超过配置的大小，进而导致编译报错。因此，建议将其设置为自适应，如下图所示：

```
1 include "kernel.config"
2 #include "busybox.config"
3 # base subset is the basic subset of all rootfs, ramdisk.
4 # - auto resize ext2 fs
5 BR2_TARGET_ROOTFS_EXT2_SIZE="AUTO"
6 # - use glibc as the basic libc
7 BR2_TOOLCHAIN_BUILDROOT_GLIBC=y
8 # - we use buildroot toolchain here, may change to custom toolchain in the future
9 BR2_TOOLCHAIN_BUILDROOT_CXX=y
```

图 8：根文件系统分区镜像大小修改位置

由于 *recovery* 和 *rootfs* 使用同一个 *base.config*，如需修改其中任意一个文件系统下的 *BR2\_TARGET\_ROOTFS\_EXT2\_SIZE* 参数，则需要将该参数复制到以下文件中：

分区	文件
<i>recovery</i>	<i>buildroot/configs/rockchip_rk356x_recovery_defconfig</i>
<i>rootfs</i>	<i>buildroot/configs/rockchip_rk3568_defconfig</i>

### 3.2.2. 其他分区镜像修改

*oem*、*userdata* 和 *persist* 等其他分区镜像需要在系统启动后挂载。随着存储文件数量的增加，可能需要修改分区镜像大小。

可以通过 *device/rockchip/rk356x/BoardConfig-rk3568-evb1-ddr4-v10.mk* 配置相关分区的大小，也可以将分区大小设置为空。当分区大小设置为空时，将根据分区中的文件大小进行自适应配置。

```
--- a/device/rockchip/rk356x/BoardConfig-rk3568-evb1-ddr4-v10.mk
+++ b/device/rockchip/rk356x/BoardConfig-rk3568-evb1-ddr4-v10.mk
@@ -52,6 +52,8 @@ export RK_USERDATA_FS_TYPE=ext4
 export RK_PERSIST_FS_TYPE=ext4
 #OEM config
 export RK_OEM_DIR=oem_normal
+#oem partition size
+export RK_OEM_PARTITION_SIZE=256M
# OEM build on buildroot
#export RK_OEM_BUILDIN_BUILDROOT=YES
#userdata config
```

图 9：其他分区镜像大小修改位置

同时也可以修改 *oem* 分区相关配置参数，如下表所示：

参数	描述
<i>RK_OEM_FS_TYPE</i>	编译文件系统类型，默认配置为 <i>ext4</i>
<i>RK_OEM_DIR</i>	编译文件系统目录，默认配置为 <i>device/rockchip/persist/persist_normal/</i>
<i>RK_OEM_PARTITION_SIZE</i>	分区大小，默认配置为自适应

### 3.3. 分区修改结果验证

编译并烧录镜像至 QSM368ZP-WF 或 SG368Z 系列模块后，验证分区是否修改成功。

#### 备注

具体编译和烧录方法，参见《Quectel\_QSM368ZP-WF&SG368Z 系列\_Linux&Ubuntu&OpenWrt\_编译&烧录指导》。

执行以下命令验证分区修改是否成功：

```
cat /proc/partitions
ls -lh /dev/block/by-name/
```

```
root@rockchip:/# cat /proc/partitions
major minor #blocks name
1 0 4096 ram0
179 0 30535680 mmcblk0
179 1 4096 mmcblk0p1
179 2 4096 mmcblk0p2
179 3 65536 mmcblk0p3
179 4 65536 mmcblk0p4
179 5 32768 mmcblk0p5
179 6 4096 mmcblk0p6
179 7 4096 mmcblk0p7
179 8 32768 mmcblk0p8
179 9 6291456 mmcblk0p9
179 10 262144 mmcblk0p10
179 11 23760864 mmcblk0p11

root@rockchip:/# ls -lh /dev/block/by-name/
total 0
lrwxrwxrwx 1 root root 15 Jan 18 12:03 backup -> ../../mmcblk0p8
lrwxrwxrwx 1 root root 15 Jan 18 12:03 boot -> ../../mmcblk0p3
lrwxrwxrwx 1 root root 15 Jan 18 12:03 misc -> ../../mmcblk0p2
lrwxrwxrwx 1 root root 15 Jan 18 12:03 nvdata1 -> ../../mmcblk0p6
lrwxrwxrwx 1 root root 15 Jan 18 12:03 nvdata2 -> ../../mmcblk0p7
lrwxrwxrwx 1 root root 16 Jan 18 12:03 oem_test -> ../../mmcblk0p10
lrwxrwxrwx 1 root root 15 Jan 18 12:03 persist -> ../../mmcblk0p5
lrwxrwxrwx 1 root root 15 Jan 18 12:03 recovery -> ../../mmcblk0p4
lrwxrwxrwx 1 root root 15 Jan 18 12:03 rootfs -> ../../mmcblk0p9
lrwxrwxrwx 1 root root 15 Jan 18 12:03 uboot -> ../../mmcblk0p1
lrwxrwxrwx 1 root root 16 Jan 18 12:03 userdata -> ../../mmcblk0p11

root@rockchip:/#
```

262144KB/1024=256MB

图 10：分区修改成功

从上图可以看出，分区名称和大小已经变更，表明分区修改成功。

## 4 分区添加

本章节以添加一个名称为 *new*，大小为 16 MB 的分区为例，介绍分区添加步骤。

### 4.1. 物理分区添加

在 Windows 系统中，使用 *parameter\_tool.exe* 工具在 *parameter-buildroot-fit.txt* 分区表文件中添加物理分区，步骤如下：

**步骤 1：** 打开 *parameter\_tool.exe* 工具，选中 *userdata* 分区，点击“向前插入”插入一个新的物理分区，如下图所示：

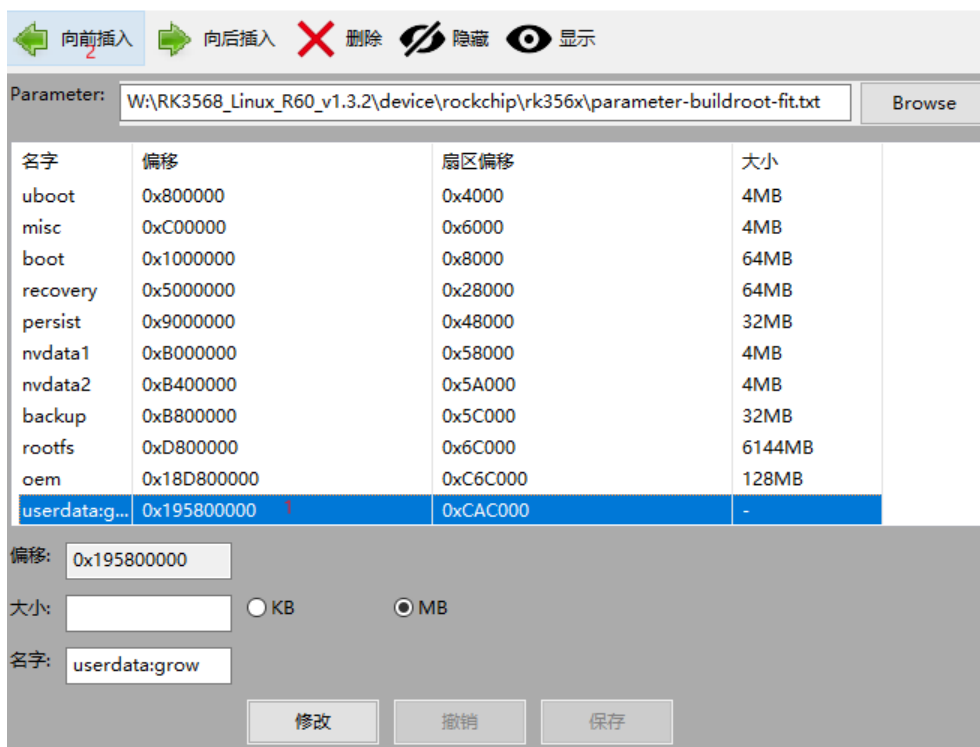


图 11：插入新物理分区



**步骤 2:** 选择新插入的分区，设置新分区大小和名称。之后，点击下图中的“修改”按钮，确认无误后，单击“保存”按钮，修改即可生效。如需取消修改，请单击“撤销”按钮。分区添加完成后，添加分区及其后面的分区的大小和偏移值将自动变更。

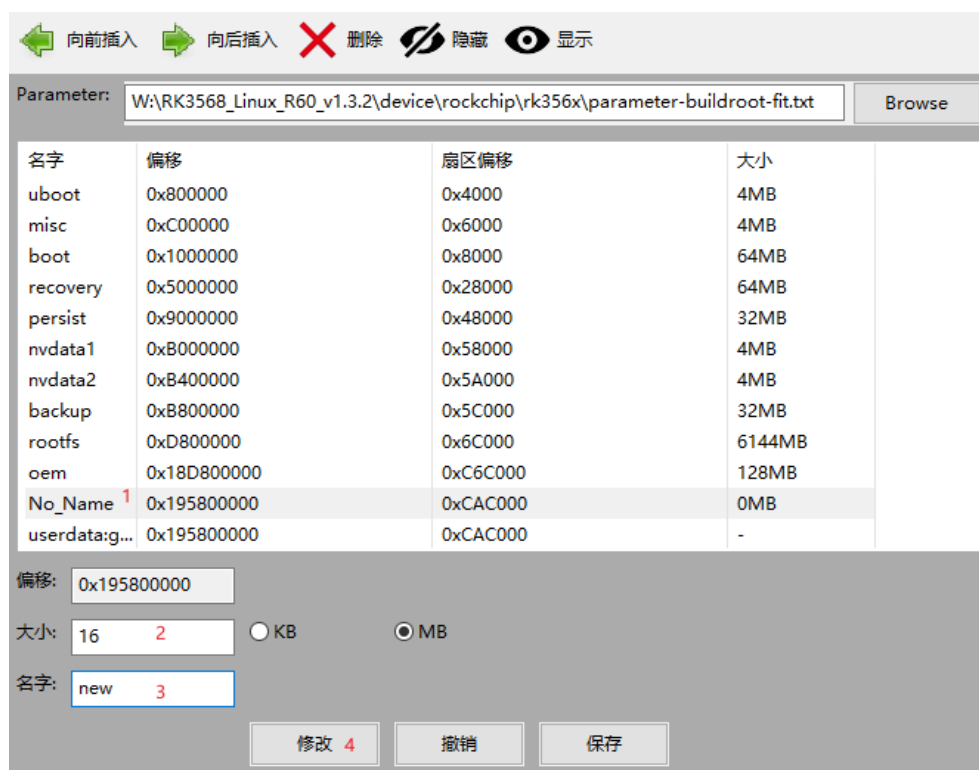


图 12: 设置新物理分区大小和名称

**步骤 3:** 检查 *parameter-buildroot-fit.txt* 文件，将会看到分区添加成功，如下图所示：

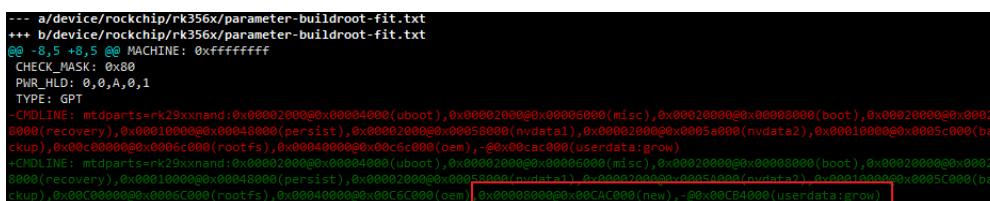


图 13: 分区添加成功

## 4.2. 分区镜像创建

### 4.2.1. Linux/Ubuntu 分区镜像创建

添加新分区镜像，首先需要了解制作系统分区镜像的文件位置，如下表所示：

配置参数	文件名称
<code>PARTITION_CONFIG</code>	<code>device/rockchip/rk356x/BoardConfig-rk3568-evb1-ddr4-v10.mk</code>
<code>PARTITION_IMAGE_PRODUCTION</code>	<code>device/rockchip/common/mkfirmware.sh</code>
<code>PARTITION_MOUNT</code>	<code>device/rockchip/common/post-build.sh</code>
<code>PARTITION_TABLE</code>	<code>device/rockchip/rk356x/parameter-buildroot-fit.txt</code>

其次需要修改 `device/rockchip/rk356x/BoardConfig-rk3568-evb1-ddr4-v10.mk`、`device/rockchip/common/mkfirmware.sh`、`device/rockchip/common/post-build.sh` 和 `device/rockchip/rk356x/parameter-buildroot-fit.txt` 文件，步骤如下：

**步骤 1：** 在 `device/rockchip/rk356x/BoardConfig-rk3568-evb1-ddr4-v10.mk` 文件中添加一个分区镜像配置，如下图所示：

```
--- a/device/rockchip/rk356x/BoardConfig-rk3568-evb1-ddr4-v10.mk
+++ b/device/rockchip/rk356x/BoardConfig-rk3568-evb1-ddr4-v10.mk
@@ -68,3 +68,10 @@ export QUECTEL_NVDATA=blank-misc.img
 export RK_DISTRO_MODULE=
 # Define pre-build script for this board
 export RK_BOARD_PRE_BUILD_SCRIPT=app-build.sh
+
+# Set new partition type, including ext2, fat
+export RK_NEW_FS_TYPE=ext4
+#new config
+export RK_NEW_DIR=new_normal
+#new partition size
+export RK_NEW_PARTITION_SIZE=16M
```

图 14：添加分区镜像配置

在 `device/rockchip/`目录下创建 `new` 分区的 `RK_NEW_DIR` 目录(下图中的 `new-normal` 目录)，并在该目录下添加此分区镜像需要安装的文件，如 `test.txt`，这些添加的文件都会被打包成 `new.img`，如下图所示：

```
$ mkdir -p device/rockchip/new/new_normal/
$ echo "Hello world!" > device/rockchip/new/new_normal/test.txt
```

图 15：添加分区文件

**步骤 2：** 添加配置，在 `device/rockchip/common/mkfirmware.sh` 中创建分区镜像，如下图所示：

```
--- a/device/rockchip/common/mkfirmware.sh
+++ b/device/rockchip/common/mkfirmware.sh
@@ -183,6 +183,7 @@ legacy_partition() {
 RK_LEGACY_PARTITIONS=" \
 $(legacy_partition oem "$RK_OEM_DIR" "$RK_OEM_FS_TYPE" "$RK_OEM_PARTITION_SIZE")
 $(legacy_partition persist "$RK_PERSIST_DIR" "$RK_PERSIST_FS_TYPE" "$RK_PERSIST_PARTITION_SIZE")
+ $(legacy_partition new "$RK_NEW_DIR" "$RK_NEW_FS_TYPE" "$RK_NEW_PARTITION_SIZE")
 $(legacy_partition userdata "$RK_USERDATA_DIR" "$RK_USERDATA_FS_TYPE" "$RK_USERDATA_PARTITION_SIZE")
"
```

图 16：添加配置创建分区镜像

### 步骤 3：自动挂载分区。

- Linux:

在 `device/rockchip/common/post-build.sh` 文件中添加开机自动挂载配置。开机后，`new` 分区将被挂载至 `/new/` 目录，如下图所示：

```
--- a/device/rockchip/common/post-build.sh
+++ b/device/rockchip/common/post-build.sh
@@ -23,6 +23,7 @@ REBOOT_WRAPPER=busybox-reboot
RK_LEGACY_PARTITIONS=" \
    ${RK_OEM_FS_TYPE:+oem:/oem:$RK_OEM_FS_TYPE}
    ${RK_PERSIST_FS_TYPE:+persist:/persist:$RK_PERSIST_FS_TYPE}
+    ${RK_NEW_FS_TYPE:+new:/new:$RK_NEW_FS_TYPE}
    ${RK_USERDATA_FS_TYPE:+userdata:/userdata:$RK_USERDATA_FS_TYPE}
"
```

图 17：添加开机自动挂载配置

- Ubuntu:

Ubuntu 中采用手动挂载，无直接配置自动挂载的位置。手动挂载详情，参见第 4.3 章。

## 4.2.2. OpenWrt 分区镜像创建

添加一个大小为 16 MB 的新物理分区（参见第 4.1 章）后，接下来需要创建新分区镜像。

OpenWrt 分区镜像添加步骤如下：

步骤 1：在 `common/images` 路径下创建一个新文件夹，命名为 `new`。

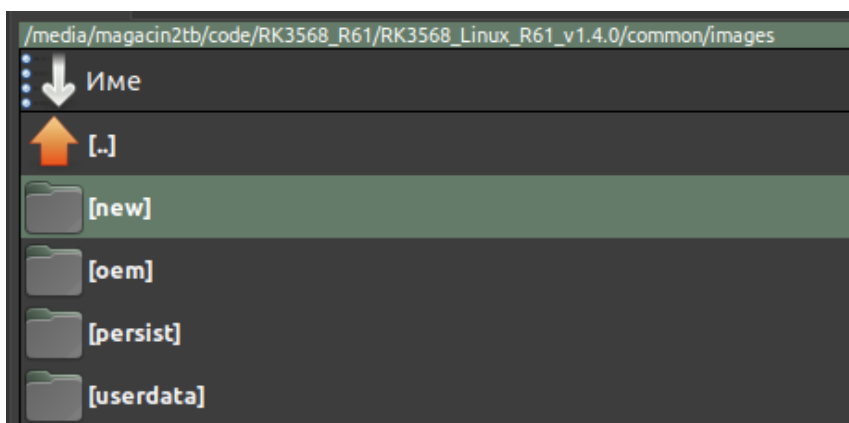


图 18：添加 new 文件夹

步骤 2: 进入 *new* 文件夹，新建一个名为 *normal* 的文件夹，并在其中添加一些文件。

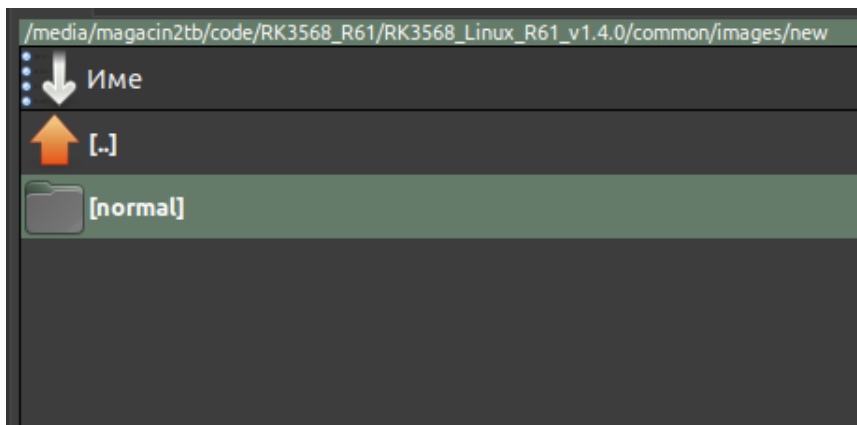


图 19: 创建 *normal* 文件夹

步骤 3: 进入 *openwrt/package/base-files/files/etc/fstab* 路径并为新分区添加如下配置:

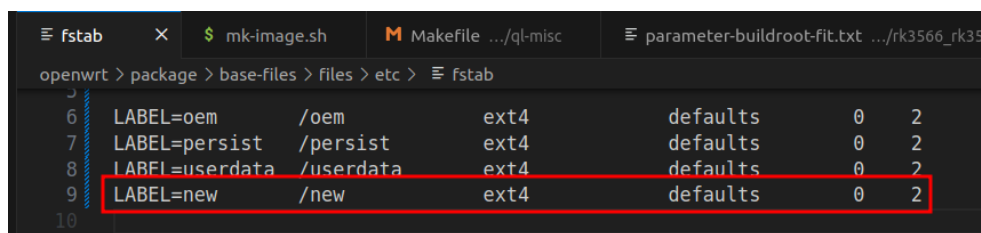


图 20: 添加新分配配置

步骤 4: 执行 **source build-quec.sh** 获取工程源码。然后执行 **make menuconfig** 命令进入 *Extra partitions* 子菜单:

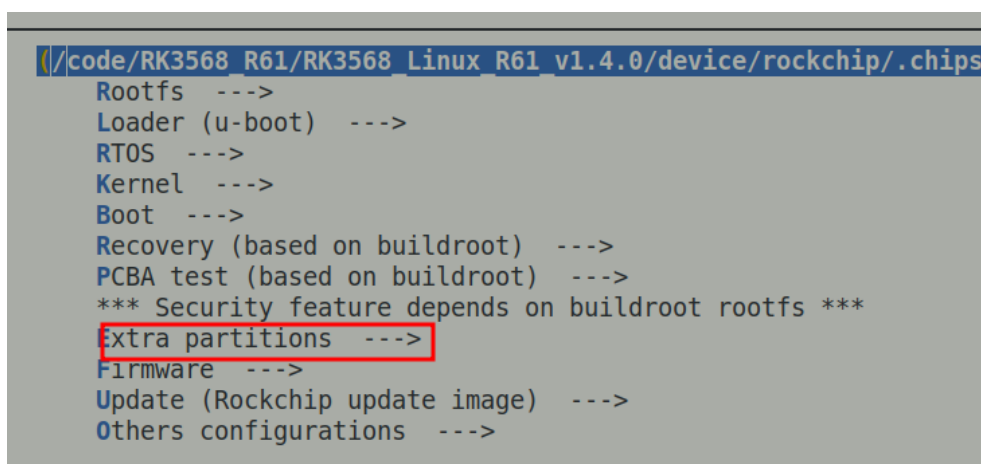


图 21: 进入 *Extra partitions* 子菜单

步骤 5: 修改 *number of extra partitions* 为 4:

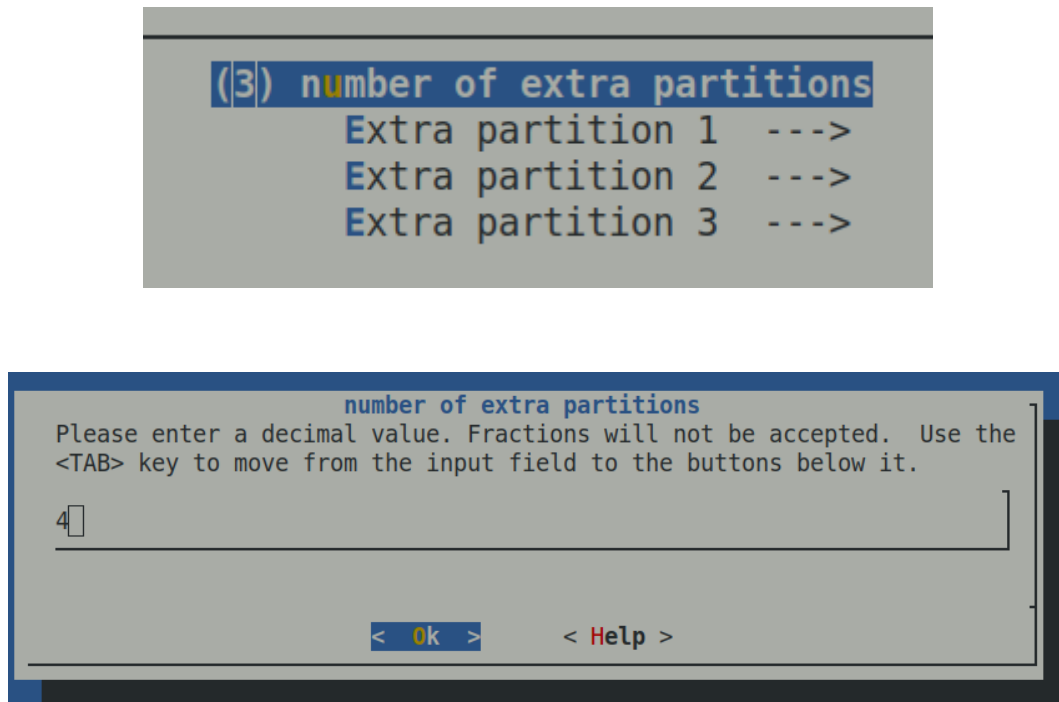


图 22: 修改 *number of extra partitions*

步骤 6: 进入新分区并修改分区相关配置:

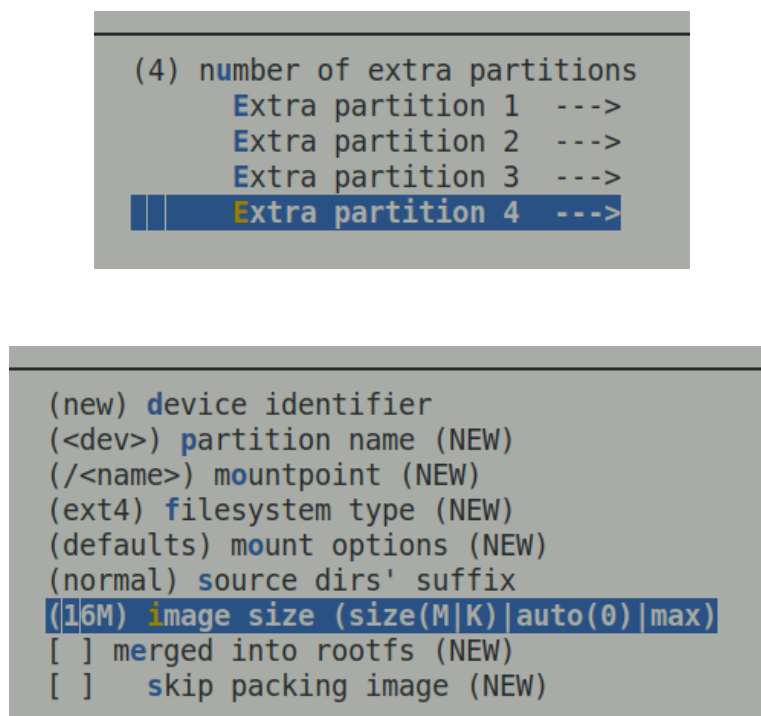


图 23: 修改分区配置

步骤 7: 保存配置:

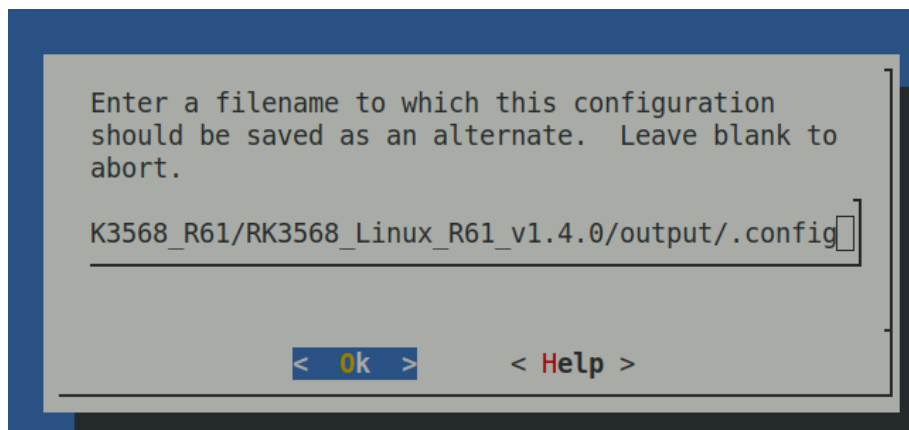


图 24: 保存配置

步骤 8: 之后退出菜单，检查 `output/.config` 文件，即可看到新添加的分区内容:

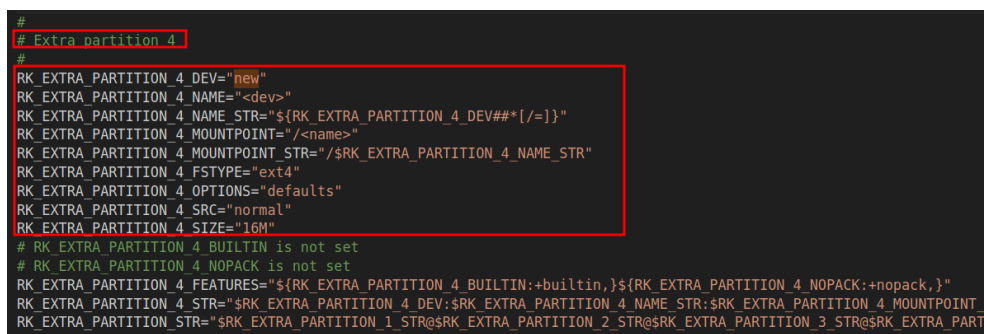


图 25: 检查 `output/.config` 新分区文件内容

步骤 9: 执行 `make` 命令进行编译（请勿执行 `build-all-image-openwrt`，否则将覆盖编译配置）。

## 4.3. 分区添加结果验证

### 4.3.1. Linux/Ubuntu 分区添加结果验证

编译并烧录镜像至 QSM368ZP-WF 或 SG368Z 系列模块后，验证分区添加是否成功。

执行以下命令验证分区添加是否成功:

```
cat /proc/partitions
ls -lh /dev/block/by-name/
```

```

root@rockchip:/# cat /proc/partitions
major minor #blocks name
1 0 4096 ram0
179 0 61079552 mmcblk0
179 1 4096 mmcblk0p1
179 2 4096 mmcblk0p2
179 3 65536 mmcblk0p3
179 4 65536 mmcblk0p4
179 5 32768 mmcblk0p5
179 6 4096 mmcblk0p6
179 7 4096 mmcblk0p7
179 8 32768 mmcblk0p8
179 9 6291456 mmcblk0p9
179 10 131072 mmcblk0p10
179 11 16384 mmcblk0p11
179 12 54419424 mmcblk0p12

root@rockchip:/# ls -lh dev/block/by-name/
total 0
lrwxrwxrwx 1 root root 15 Aug 24 03:11 backup -> ../../mmcblk0p8
lrwxrwxrwx 1 root root 15 Aug 24 03:11 boot -> ../../mmcblk0p3
lrwxrwxrwx 1 root root 15 Aug 24 03:11 misc -> ../../mmcblk0p2
lrwxrwxrwx 1 root root 16 Aug 24 03:11 new -> ../../mmcblk0p11
lrwxrwxrwx 1 root root 15 Aug 24 03:11 nvdata1 -> ../../mmcblk0p6
lrwxrwxrwx 1 root root 15 Aug 24 03:11 nvdata2 -> ../../mmcblk0p7
lrwxrwxrwx 1 root root 16 Aug 24 03:11 oem -> ../../mmcblk0p10
lrwxrwxrwx 1 root root 15 Aug 24 03:11 persist -> ../../mmcblk0p5
lrwxrwxrwx 1 root root 15 Aug 24 03:11 recovery -> ../../mmcblk0p4
lrwxrwxrwx 1 root root 15 Aug 24 03:11 rootfs -> ../../mmcblk0p9
lrwxrwxrwx 1 root root 15 Aug 24 03:11 uboot -> ../../mmcblk0p1
lrwxrwxrwx 1 root root 16 Aug 24 03:11 userdata -> ../../mmcblk0p12

```

图 26：添加物理分区成功

由于 Ubuntu 没有配置自动挂载分区，需要执行如下命令挂载新分区：

```

adb shell
mount -o remount,rw /
mkdir /new
mount -t ext4 /dev/block/bootdevice/by-name/new /new

```

执行以下命令验证分区镜像中的文件是否添加成功：

```

df -h
cat /new/test.txt

```

```

root@rockchip:/# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        944M  269M  623M  31% /
devtmpfs        3.8G   0  3.8G   0% /dev
tmpfs           3.8G  192K  3.8G   1% /tmp
tmpfs           3.8G  372K  3.8G   1% /run
tmpfs           3.8G   0  3.8G   0% /dev/shm
/dev/mmcblk0p10  15M   12M  1.6M  89% /oem
/dev/mmcblk0p5   26M   17K  23M   1% /persist
/dev/mmcblk0p11  14M   15K  13M   1% /new
/dev/mmcblk0p12  3.0M  291K  2.4M  11% /userdata
root@rockchip:/# cat /new/test.txt
Hello world!

```

图 27：分区镜像文件添加成功

#### 4.3.2. OpenWrt 分区添加结果验证

编译并烧录镜像至 QSM368ZP-WF 或 SG368Z 系列模块后，验证分区添加是否成功。

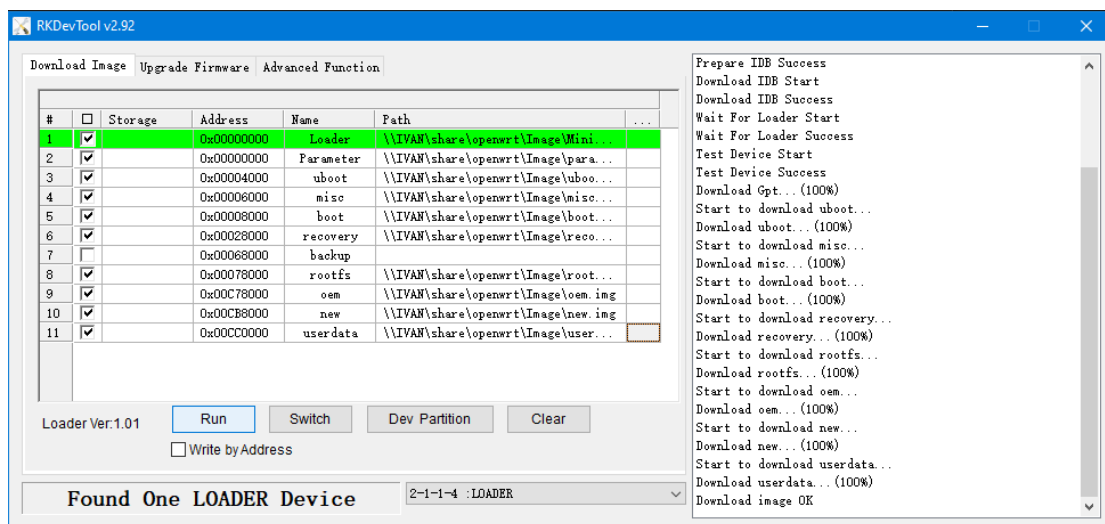


图 28：使用 RKDevTool 工具烧录固件

烧录成功后，开机并执行 `./adb shell` 命令，将会看到如下内容：

```
PS C:\Users\Ivan\Desktop\platform-tools> ./adb shell
* daemon not running; starting now at tcp:5037
* daemon started successfully

BusyBox v1.35.0 (2023-11-25 18:18:57 UTC) built-in shell (ash)

 _ _ _ _ _
|_ _ _ _ _|_ _ _ _ _|_ _ _ _ _|_ _ _ _ _|_ _ _ _ _|_ _ _ _ _|_ _ _ _ _|_ _ _ _ _|
|_| WIRELESS FREEDOM

-----
OpenWrt 22.03.6, r20265-f85a79bcb4
-----
=== WARNING! =====
There is no root password defined on this device!
Use the "passwd" command to set up a new password
in order to prevent unauthorized SSH logins.
-----
root@OpenWrt:/#
```

图 29：OpenWrt 环境

由于使能了自动挂载，且配置了 `/etc/fstab`，此时可以检查挂载点：

```
root@OpenWrt:/# mount
/dev/mmcblk0p6 on / type ext4 (rw,noatime)
devtmpfs on /dev type devtmpfs (rw,relatime,size=477064k,nr_inodes=119266,mode=755)
proc on /proc type proc (rw,nosuid,nodev,noexec,noatime)
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,noatime)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev,noatime)
tmpfs on /dev type tmpfs (rw,nosuid,noexec,noatime,size=512k,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,noatime,mode=600,ptmxmode=000)
debugfs on /sys/kernel/debug type debugfs (rw,noatime)
pstore on /sys/fs/pstore type pstore (rw,noatime)
none on /sys/kernel/config type configfs (rw,relatime)
/dev/mmcblk0p7 on /oem type ext4 (rw,relatime)
/dev/mmcblk0p9 on /userdata type ext4 (rw,relatime)
/dev/mmcblk0p8 on /new type ext4 (rw,relatime)
adb on /dev/usb-lun0 type usbfs (rw,relatime)
root@OpenWrt:/#
```

图 30：检查挂载点



可以看到新分区已挂载成功，执行如下图 **ls -al** 命令检查是否合入了正确的文件。若合入了正确文件，将显示如下图所示内容：

```
root@OpenWrt:/new# ls -al
drwxr-xr-x  3 root    root      4096 Aug  4 09:00 .
drwxr-xr-x 22 root    root      4096 Nov 25 2023 ..
-rw-r--r--  1 root    root         0 Aug  4 09:00 .resized
-rw-r--r--  1 root    root    87402 Aug 15 2024 200frames_count.h264
-rw-r--r--  1 root    root  512017 Aug 15 2024 SampleJPGImage_500kbmb.jpg
-rw-r--r--  1 root    root  5253880 Aug 15 2024 SampleVideo_1280x720_5mb.mp4
-rw-r--r--  1 root    root   81582 Aug 15 2024 belle-nuit-testchart-1080p.png
drwx----- 2 root    root    16384 Aug 26 2024 lost+found
-rw-r--r--  1 root    root   101760 Aug 15 2024 piano2-CoolEdit.mp3
-rw-r--r--  1 root    root   96092 Aug 15 2024 retroarch.cfg
root@OpenWrt:/new#
```

图 31：new 分区文件内容